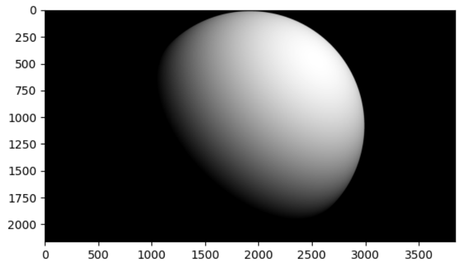
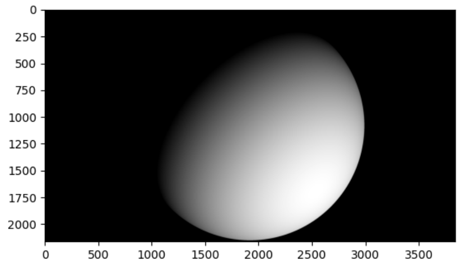
Q1 (a)

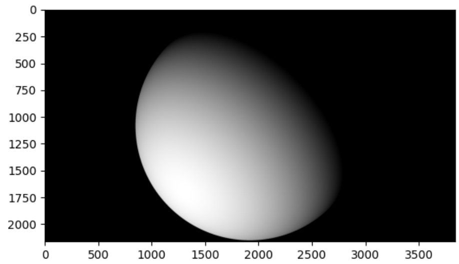
The surface radiance is given by $L = \frac{\rho_d}{\pi} I \cos\theta_i$, where $\cos\theta_i = n \cdot s$ because n and s are unit vectors so $n \cdot s = \cos\theta_i$.
The projected area is the original area times $\cos\theta_i$. In the equation, it is $I \cos\theta_i$.
The viewing direction does not matter because under Lambertian theory, surface appears equally bright from all directions.

Q1 (b)

Rendering results are shown below.







Q1 (c)

The screenshot for loadData function is shown below (see Figure 1).

```python
def loadData(path = "../data/"):
    """
    Question 1 (c)
    Load data from the path given. The images are stored as input_n.tif
    for n = {1...7}. The source lighting directions are stored in
    sources.mat.

    Paramters
    ---------
    path: str
        Path of the data directory
    Returns
    -------
    I : numpy.ndarray
        The 7 x P matrix of vectorized images

    L : numpy.ndarray
        The 3 x 7 matrix of lighting directions

    s: tuple
        Image shape

    """
    I = None
    L = None
    s = None
    # read images
    for i in range(7):
        input_path = path + f'input_{i+1}.tif'
        img = cv2.imread(input_path, cv2.IMREAD_UNCHANGED)
        h, w = img.shape[:2]
        if I is None:
            I = np.zeros((7, h*w))
        if s is None:
            s = (h, w)
        img = rgb2xyz(img) # RGB -> XYZ
        luminance = img[:, :, 1] # luminance channel
        I[i,] = luminance.reshape(-1)

    # lighting directions
    L = np.load(path + 'sources.npy')
    L = L.T
    return I, L, s
```

Figure 1: Screenshot of loadData function.

Q1 (d)

We have three unknown variables in b, so we need 3 different light sources, which are not in the same plane, to estimate the normals. Thus, the matrix L^T has rank 3. Since $I = L^T B$, we can know that the rank of I should also be 3.
The singular values are
72.40617702, 12.00738171, 8.42621836, 2.23003141, 1.51029184, 1.17968677, 0.84463311
No, the singular values do not agree with the rank-3 requirement. Because the image intensities captured might contain noise, so there are more independent light sources required to estimate the normals.

Q1 (e)

A general linear system can be written as $Ax = b$. In our case, we have $I = L^T B$, where we know I and L.
Since L^T is not a square matrix, we multiply a L on both sides:

$$LI = (LL^T)B$$

Therefore, $y = LI$ and $A = LL^T$.

Q1 (f)

The albedo is shown in Figure 2. We can see some regions around the nose and ears that are brighter than it should be. Because it may capture regions that have multiple light sources, which is against the n-dot-l model.

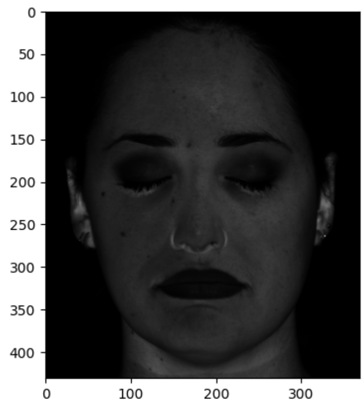


Figure 2: Albedo

The surface normals are shown in Figure 3. It matches the curvature of the face.

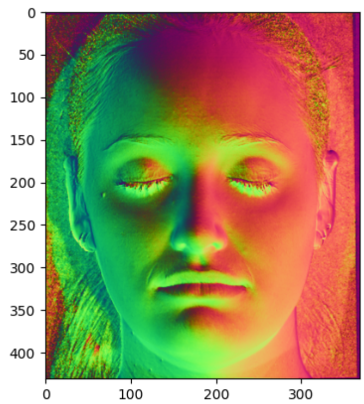


Figure 3: Surface normals.

Q1 (g)

Consider the 3D depth equation $z - f(x, y) = 0$. The surface normal can be expressed as gradients in each location: $n = (-\frac{\partial f}{\partial x}, -\frac{\partial f}{\partial y}, 1)$.
The normals can also be written as: $n = (n_1, n_2, n_3)$. Normalize it by dividing by n_3, we can obtain $n = (\frac{n_1}{n_3}, \frac{n_2}{n_3}, 1)$.
Thus, we have $\frac{\partial f}{\partial x} = -\frac{n_1}{n_3}$ and $\frac{\partial f}{\partial y} = -\frac{n_2}{n_3}$.

Q1 (h)

We can know the x and y gradient matrix: $g_x = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$ and $g_y = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \\ 4 & 4 & 4 & 4 \end{pmatrix}$.
First, let's use g_x to construct the first row. Since we know $g(0,0) = 1$, then $g(1,0) = g(0,0) + g_x(0,0) = 2$. Similarly, we can obtain the first row as $\begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix}$.
Then we use g_y to construct the rest of g. $g(0,1) = g(0,0) + g_y(0,0) = 5$.
Similarly, we can obtain the rest of the matrix, which is $\begin{pmatrix} 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$.

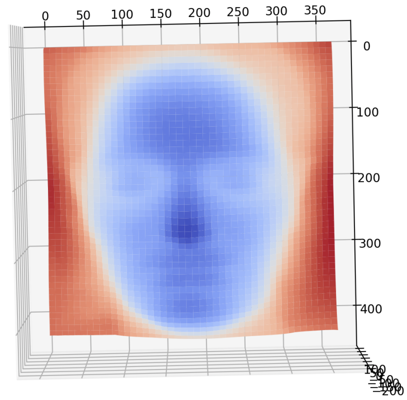
Therefore, the full matrix $g = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$.

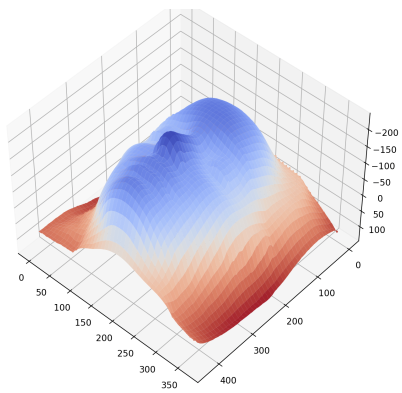
Then we consider first using g_y to construct g and then g_x.
The result $g = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$.
The results are the same.

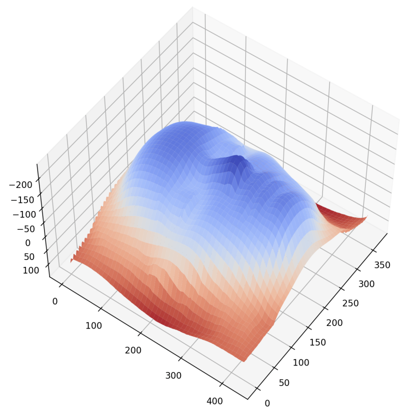
By modifying the gradient values, we can make g_x and g_y non-integrable. For example, if $g_x(0,1) = 2$. Then if we use g_x to calculate $g(1,1)$, we will obtain $g(1,1) = g_x(0,1) + g(0,1) = 7$.
If we use g_y to calculate $g(1,1)$, we will obtain $g(1,1) = g_y(1,0) + g(1,0) = 6$. And they are not equal.
The gradients estimated in (g) may not be integrable. The reason might be the pixel lengths of the horizontal and vertical directions are not the same.

Q1 (i)

Three viewpoints of the reconstructed shape are shown below.







Q2 (a)

For $I = L^T B$, we can apply SVD to I. Then we can obtain U, Σ, V^T.
Because of the rank constraint, we only keep the top 3 singular values in Σ and set all others to 0. That is equivalent to the case where we extract the first 3 columns of U, the first 3 singular values of Σ and the first 3 rows of V^T.

Q2 (b)

The albedos and normals are shown below in Figure 4.

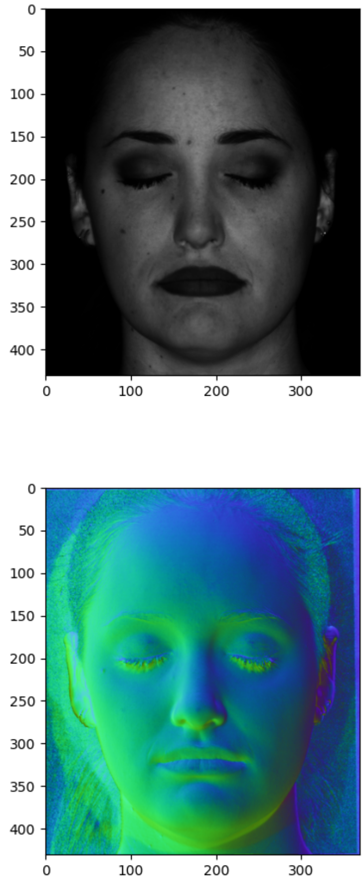


Figure 4: Albedos and Normals

Q2 (c)

The groundtruth light and the estimated light are shown below.

```
Groundtruth Light: [[-0.1418  0.1215 -0.069   0.067  -0.1627  0.      0.1478]
 [-0.1804 -0.2026 -0.0345 -0.0402  0.122   0.1194  0.1209]
 [-0.9267 -0.9717 -0.838  -0.9772 -0.979  -0.9648 -0.9713]]
Estimated Light: [[-0.81724946 -0.83704223 -0.96462606 -0.98651718 -0.83715792 -0.95690046
  -0.85380093]
 [ 0.2578772  -0.50172823  0.19976913 -0.16355506  0.5424538   0.13324687
  -0.19929221]
 [ 0.51536654  0.21824092  0.17201411 -0.00579668 -0.07007493 -0.25804413
  -0.48094344]]
```

Figure 5: Groundtruth light and estimated light by SVD.

The $\hat{L}$ estimated and ground truth L_0 are quite different.
For now, I use $L^T = U\Sigma^{\frac{1}{2}}$ and $B = \Sigma^{\frac{1}{2}}V^T$.
I can also do $L^T = 2U\Sigma^{\frac{1}{2}}$ and $B = \frac{1}{2}\Sigma^{\frac{1}{2}}V^T$. Because the scale does not matter for surface normals, the rendered images will not change.

Q2 (d)

The reconstructed surface is shown below. It does not look like a face.

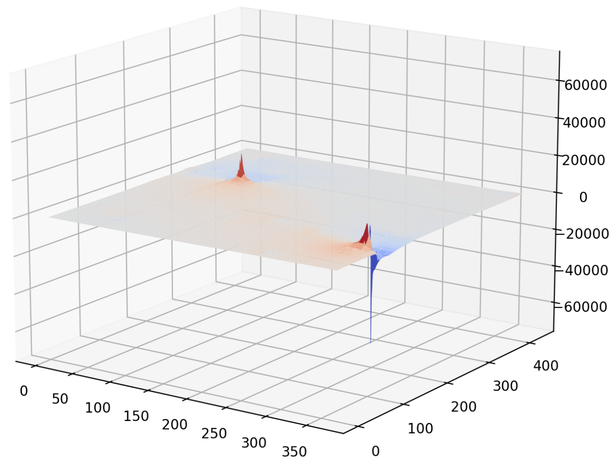
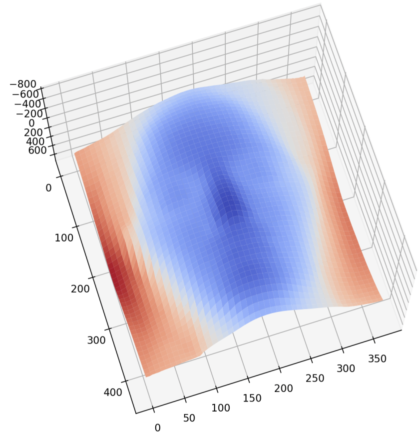
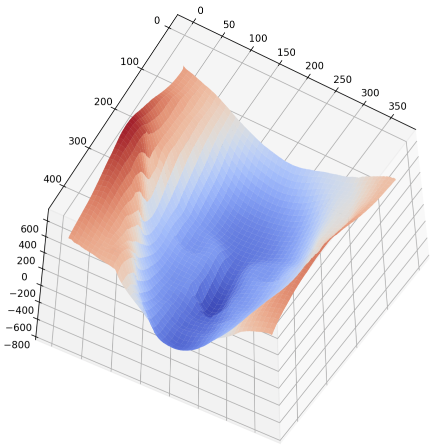


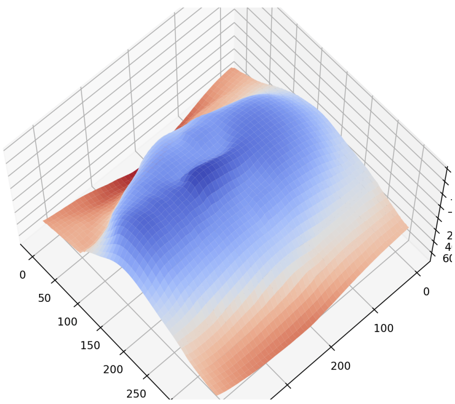
Figure 6: Reconstructed surface.

Q2 (e)

Three viewpoints of the reconstructed shape are shown below.
From the reconstructed face, it looks like the output of calibrated photometric stereo. But there are still many parts that are different. For example, the non-face regions are not flat.







Q2 (f)

Let's start with an identity matrix G, where $\mu = 0, v = 0, \lambda = 1$.
In the first set of experiment, let's vary μ and keep v and λ unchanged, i.e., $v = 0, \lambda = 1$ (see Figure 7 and 8).
We can see the surface is tilted to the positive direction of x axis.

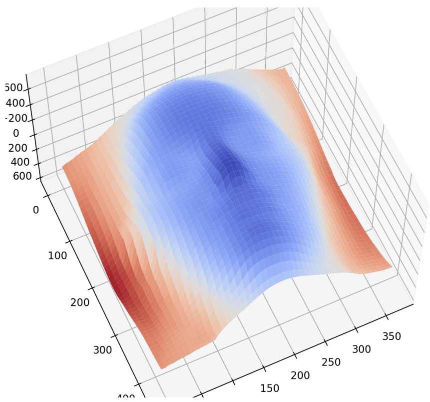


Figure 7: $\mu = 1$

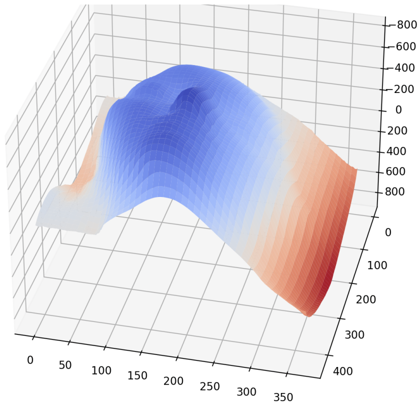


Figure 8: $\mu = 10$

In the second set of experiment, let's vary v and keep μ and λ unchanged, i.e., $\mu = 0, \lambda = 1$ (see Figure 9 and 10).
We can see the surface is tilted to the negative direction of x axis.

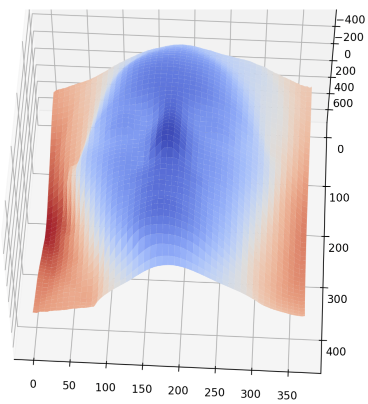


Figure 9: $v = 1$

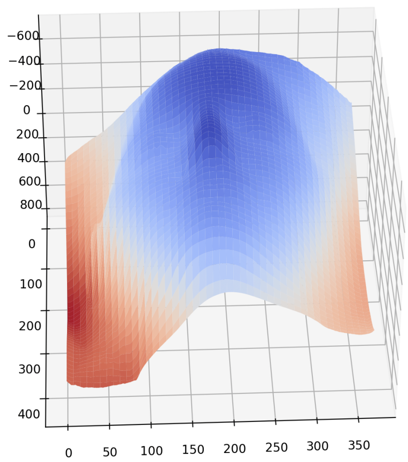


Figure 10: $v = 10$

In the third set of experiment, let's vary λ and keep μ and v unchanged, i.e., $\mu = 0, v = 0$ (see Figure 11 and 12).
We can see that λ controls the flatness of the surface. The smaller λ becomes, the flatter the surface will be.

The bas-relief ambiguity is so named because it's difficult to distinguish between changes in the shape of a surface and changes in the lighting direction based on

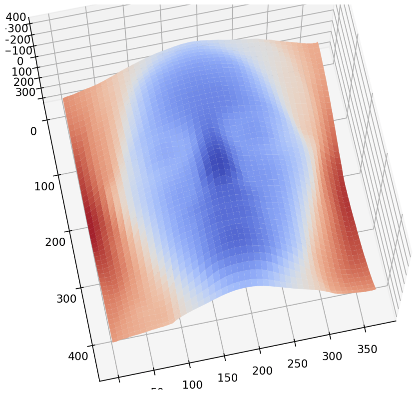


Figure 11: $\lambda = 0.5$

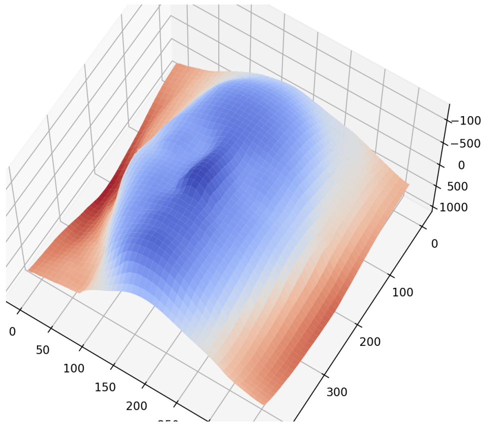


Figure 12: $\lambda = 2$

shading information alone. This is because a slight change in the lighting direction can produce a shading pattern similar to a change in the shape of the surface. For example, a smoothly curved surface can appear to have a more pronounced curvature simply by altering the light direction.

Q2 (g)

To make the estimated surface as flat as possible, I will set a small value for λ. μ and v do not really matter for the flatness of surface.

Q2 (h)

No, acquiring more pictures from more lighting directions will not help resolve the ambiguity. We still have the same ambiguity when factorizing the intensity matrix.